



⑯ BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

⑯ ⑫ **Offenlegungsschrift**  
⑯ ⑯ **DE 198 10 814 A 1**

⑯ ⑯ Int. Cl. 6:  
**G 06 F 11/28**

DE 198 10 814 A 1

⑯ ⑯ Aktenzeichen: 198 10 814.1  
⑯ ⑯ Anmeldetag: 12. 3. 98  
⑯ ⑯ Offenlegungstag: 16. 9. 99

⑯ ⑯ Anmelder:  
Telefonaktiebolaget L M Ericsson (publ),  
Stockholm, SE  
⑯ ⑯ Vertreter:  
HOFFMANN · EITLE, 81925 München

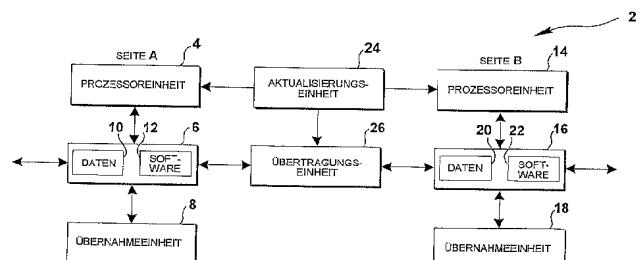
⑯ ⑯ Erfinder:  
Gard, Bengt Erik Ingemar, Tullinge, SE; Kling,  
Lars-Örjan, Södertälje, SE; Johnsson, Sten Edvard,  
Bandhagen, SE  
⑯ ⑯ Entgegenhaltungen:  
DE 41 34 207 C1  
DE 44 29 969 A1  
US 51 55 837

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

Prüfungsantrag gem. § 44 PatG ist gestellt

⑯ Zustandskopierverfahren für eine Software-Aktualisierung

⑯ Zum Bereitstellen einer Vorhergehensweise für die Software-Aktualisierung mit skalierbarer Störung wird ein Zustandskopierverfahren für ein Rechensystem mit zumindest zwei logischen Partitionen vorgeschlagen. Ein Zustand einer neuen Software in einer Standby-Partition (6, 16) wird dem Zustand einer alten Software in einer ausführenden Partition bei Fortführung der alten Software angeglichen. Die Daten werden zwischen der ausführenden Partition und der Standby-Partition in skalierbarer Weise übertragen, und sobald derselbe Zustand für die Standby-Partition (6, 16) und die ausführende Partition (16, 6) erzielt ist, wird die Durchführung zu der neuen Software umgeschaltet. Dies ermöglicht die Skalierung des Umfangs der Störung aufgrund der Softwareaktualisierung auf einen zulässigen Umfang.



DE 198 10 814 A 1

## Beschreibung

## Gebiet der Erfindung

Die vorliegende Erfindung betrifft die Aktualisierung von Software und insbesondere eine Funktionsänderung in Computer basierten Systemen mit häufiger Aktualisierung aufgrund neu hinzugefügter Funktionalität und/oder einer Korrektur von Fehlern.

## Hintergrund der Erfindung

Die Entwicklung von Datenverarbeitungsanlagen und der Software-Technologie führt zu einem zunehmenden Bedarf für Verfahren zum Aktualisieren installierter Software.

Die übliche Vorgehensweise besteht im Stoppen der Durchführung der installierten Software, dem Laden der neuen Software und dem anschließenden Starten der neuen Software. Bei dieser Vorgehensweise werden keine internen Daten zwischen der alten und der neuen Software übertragen. Weiterhin gehen bei diesem Verfahren eingerichtete Dienste verloren, und der Dienst wird während dem Laden und dem Starten der neuen Software vollständig gestoppt. Momentan wird dieses Verfahren typischerweise für Workstations und Personal-Computer eingesetzt.

Eine weitere Vorgehensweise zum Aktualisieren von Software ist in EP-A-0 201 281 beschrieben. Jedoch ermöglicht diese Lösung keine störungsfreie Datenaktualisierungsfunktion, da jede erforderliche Umsetzung von Datenmeldungen durch die neu installierte Software selbst während deren Anlauf durchgeführt wird.

Ferner wird in US-A-5 155 837 vorgeschlagen, die Eingabe von Daten für neue Dienste zu der neuen Software in einem ersten Schritt umzuschalten. Ferner wird bei Abschluß eines aktivierten Dienstes in der alten Software die Ausgabe der Daten von dem Dienst von der alten Version zu der neuen Version umgeschaltet. Jedoch kann mit dieser Lösung lediglich Software gehandhabt werden, die Dienste mit einer sehr kurzen Dauer implementiert, da die Software nach der alten Version erst bearbeitet werden muß, bevor die neue Softwareversion voll betriebsbereit ist.

Demnach existiert bei allen bekannten Vorgehensweisen eine bestimmte Störung des Systembetriebs im Fall einer Softwareaktualisierung. Diese Störung liegt zwischen einem vollständigen Herunterfahren des Systems über Stunden oder möglicherweise Tage hinweg und einer kurzen Unterbrechung möglicherweise lediglich im Hinblick auf einige begrenzte Teile der gesamten Systemfunktionalität während einiger Sekunden. Gegebenenfalls wird überhaupt keine Störung wahrgenommen, obgleich dies bei real existierenden Systemen, wie Telekommunikationsvermittlungen, nicht der Fall ist.

## Zusammenfassung der Erfindung

Im Hinblick auf die obigen Ausführungen besteht eine Aufgabe der Erfindung in der Schaffung einer Vorgehensweise für die Aktualisierung von Software, die mit minimaler Störung realisierbar und auf nahezu keine Störung skalierbar ist.

Gemäß der vorliegenden Erfindung wird diese Aufgabe gelöst durch ein Software-Bearbeitungsgerät vom Typ mit Aktualisierungs-Funktionalität, enthaltend eine Speichervorrichtung, unterteilt in eine ausführende Speicherpartitionierungsvorrichtung zum Speichern einer ersten Gruppe von Softwaremodulen und zugeordneter Daten, und eine Standby-Speicherpartitionierungsvorrichtung zum Speichern einer zweiten Gruppe von Softwaremodulen und zugeordneter

Daten, eine Aktualisierungs-Steuervorrichtung zum Aktualisieren eines Zustands der neuen Software in der Standby-Speicherpartitionierungsvorrichtung gemäß dem Zustand der alten Software in der ausführenden Speicherpartitionierungsvorrichtung während der Fortführung der Durchführung der alten Software, und eine Übertragungsvorrichtung für die skalierbare Übertragung von Daten von der ausführenden Speicherpartitionierungsvorrichtung zu der Standby-Speicherpartitionierungsvorrichtung.

10 Demnach wird das aktualisierende System in zwei logische Partitionen aufgeteilt. Diese Partitionen können, müssen jedoch nicht unter Einsatz eines Prozessorspaars implementiert sein. Hierbei enthält gemäß der Erfindung eine als ausführende Partition bezeichnete Partition die alte Software, die eine normale Softwarebearbeitung durchführt. Ferner wird die neue Software in die andere Partition geladen, die als Standby- bzw. Bereitschaftspartition bezeichnet wird, ohne Störung der Durchführung der Betriebssoftware. Die Software in der Standby-Partition wird in denselben Zustand gebracht wie die Software in der ausführenden Partition, so daß die neue Software in der Standby-Partition die normale Programmdurchführung ohne jegliche Störung übernehmen kann. Dies wird durch Kopieren von Daten von der Betriebspartition durchgeführt. Da die alte Software und die neue Software nicht identisch sind, kann der Fall eintreten, daß Daten in eine für die neue Software geeignete Darstellung umzusetzen sind. Gemäß der vorliegenden Erfindung erfolgt diese Umsetzung parallel zu und ohne Störung der Betriebssoftware, die fortlaufend in der Betriebspartition durchgeführt wird.

Weiterhin ist es in dem Fall, in dem die Übertragung sämtlicher Daten an der alten Software nicht mehr praktikabel ist, gemäß der vorliegenden Erfindung möglich, Daten teilweise von der alten Software zu übertragen. Dies ermöglicht die Skalierung des Umfangs der Störung aufgrund der Softwareaktualisierung in dem System.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung enthält die Aktualisierungs-Steuervorrichtung ferner eine Umschaltvorrichtung und eine Zustandsvergleichsvorrichtung zum Umschalten der Durchführung zu der neuen Software, sobald derselbe Zustand für die Standby-Partition und die Betriebspartition durch die Zustandsvergleichsvorrichtung detektiert ist.

Somit erfordert gemäß der vorliegenden Erfindung das Umschalten von der alten Software zu der neuen Software, daß der gesamte Zustand, wie er sich in sämtlichen Daten der alten Software widerspielt, zu der neuen Software kopiert und gleichzeitig, falls erforderlich, umgesetzt wird. Demnach ist es gemäß der vorliegenden Erfindung möglich, den Betrieb mit der neuen Software ohne jegliche Störung fortzusetzen. Ferner können in dem Fall, in dem Daten bei Programmen der alten Software vorliegen, die zum Zeitpunkt des Umschaltens nicht bearbeitet werden, diese vor dem Start der neuen Software kopiert und, falls erforderlich, umgesetzt werden.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung ist jeder Speicherpartition zumindest eine Übernahmeverrichtung zum Durchführen von Vorgabeaktionen für den Fall zugeordnet, daß Daten im Zusammenhang mit der alten Software lediglich teilweise übertragen werden, so daß eine spezielle Übernahmeverrichtung unmittelbar nach dem Umschalten aktiviert wird.

Hier wird die spezielle Übernahmeverrichtung unmittelbar nach dem Umschalten aktiviert, und sie führt Vorgabeaktionen aus, die keine vollständige Eingabe von Daten erfordern. Obgleich in diesem Fall eine Störung in einem bestimmten Umfang auftreten kann, und zwar in dem Umfang, wie Daten der alten Software fehlen, ermöglicht die vorlie-

gende Erfindung durch die Miteinbeziehung von Vorgabekonfigurationen jedoch eine Skalierung, die durch den Umfang der zulässigen Störung bestimmt ist.

Gemäß einer weiteren zusätzlichen Ausführungsform der vorliegenden Erfindung weist die Aktualisierungs-Steuervorrichtung eine Fortsetzung der alten Software in der Betriebspartition dann an, wenn eine Fehlersituation vor dem Umschalten auftritt. Ebenfalls führt sie eine Rück-Umschaltung so durch, daß die Partition mit der alten Software erneut die Betriebspartition wird, und zwar in dem Fall, in dem ein Fehler während der Durchführung der neuen Software nach dem Umschalten auftritt.

Tritt ein Fehler vor dem Umschalten auf, so wird die Aktualisierung der Software abgeschlossen, und die normale Softwaredurchführung wird mit der alten Software in der Betriebspartition fortgesetzt. Tritt anderenfalls ein Fehler während der Durchführung der neuen Software nach dem Umschalten auf, so wird ein Rück-Umschalten so durchgeführt, daß die Partition mit der alten Software erneut die Betriebspartition wird. Hierbei kann die Rück-Umschaltung Schritte zum Kopieren von Daten und, falls erforderlich, für ein Umsetzen aufweisen, in derselben Weise wie die Umschaltprozedur. Demnach kann die Rück-Umschaltprozedur ebenfalls mit begrenzter oder ohne jede Störung durchgeführt werden. Alternativ kann sie ohne jegliches Datenkopieren und -umsetzen durch Anstoßen einer Rücksetzprozedur bzw. Recovery-Prozedur rückgeführt werden, was typischerweise zu einem gewissen Umfang von Störungen führt.

Ferner wird gemäß der vorliegenden Erfindung die oben genannte Aufgabe ebenfalls gelöst durch ein Zustandskopierverfahren für ein Rechensystem mit zumindest zwei Logikpartitionen, enthaltend die Schritte Aktualisieren eines Zustands der neuen Software in einer Standby-Partitioniervorrichtung an den Zustand der alten Software in einer ausführenden Partitioniervorrichtung bei Fortführung der Durchführung der alten Software, Umschalten zum Durchführen der neuen Software, sobald derselbe Zustand für die Standby-Partitioniervorrichtung und die ausführende Partitioniervorrichtung erzielt wird.

Demnach ist es durch Einsatz des Verfahrens gemäß der vorliegenden Erfindung möglich, eine hochwirksame und störungsfreie Aktualisierung von Software selbst dann zu erzielen, wenn alte Software vorliegt, die Dienste mit langer Dauer handhabt.

Gemäß einer bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens enthält der Aktualisierungsschritt ferner einen Initialisierungsteilschritt, der parallel zu und ohne Störung der in der Betriebspartition ablaufenden alten Software durchgeführt wird.

Somit folgen auf die Aktualisierung der neuen Software gegebenenfalls die Initialisierungsroutine. Obgleich dies teilweise früher erfolgen kann, beispielsweise unmittelbar nach dem Laden der neuen Software, ist ein Teil dieser Initialisierung durch Daten der alten Software bestimmt und kann nicht vorab durchgeführt werden. Die Initialisierung der neuen Software erfolgt parallel mit minimaler Störung der üblichen Betriebsssoftware, die in der Betriebspartition weiter ausgeführt wird. Da sich der Zustand der Betriebspartition fortlaufend verändert, ist der störungsfreie Aktualisungsprozeß gemäß der vorliegenden Erfindung ebenfalls fortlaufend parallel mit der Initialisierung durchzuführen.

Gemäß einer anderen weiteren bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens wird der Aktualisierungsschritt wiederholt als Hintergrundprozeß bis zum Umschalten zu der neuen Software durchgeführt, zum Berücksichtigen des sich verändernden Zustand in der Betriebspartition. Ist der vollständige Zustand, wie es sich an-

hand sämtlicher Daten der alten Software darstellt, zu der neuen Software kopiert und, falls erforderlich, umgesetzt, so ist es möglich, den Betrieb der neuen Software ohne irgend eine Störung fortzusetzen. Erfolgt ein Datenaustausch zwischen Programmen der alten Software mit Daten, die zum Zeitpunkt des Umschaltens nicht bearbeitet werden, so müssen diese auch kopiert und, falls erforderlich, umgesetzt werden.

Gemäß einer weiteren bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens werden Daten im Zusammenhang mit alter Software lediglich teilweise übertragen, und ein spezieller Übernahmeschritt wird unmittelbar nach dem Umschalten durchgeführt, zum Durchführen von Vorgabeaktionen, die keine vollständige Eingabe von Daten erfordern. In diesem Fall kann eine gewisse Störung auftreten. Der Umfang der Störung ist davon abhängig, wieviele Daten der alten Software nicht vorliegen. Vorteilhafterweise kann prinzipiell eine Skalierung in dem Umfang erfolgen, der als geeignet angesehen wird.

Ferner wird gemäß der vorliegenden Erfindung ein Zustandskopierverfahren geschaffen, und zwar für ein verteiltes Rechensystem mit mindestens einem Hauptprozessor und zumindest einem Remote-Prozessor, das die Schritte Aktualisieren der neuen Software in einer ersten/Standby-Speicherpartition des Remote-Prozessors enthält, sowie ein Aktualisieren des Zustands der neuen Software zum Erzielen eines Abgleichs mit dem Zustand in dem Hauptprozessor bei Fortführung der Durchführung der Software in dem Hauptprozessor, und ein Umschalten der Durchführung der Software von dem Remote-Prozessor zu der neuen Software, sobald ein Abgleich mit dem Zustand in dem Hauptprozessor erzielt ist.

Dieses modifizierte Verfahren gemäß der vorliegenden Erfindung ermöglicht das Erzielen einer Aktualisierung von Softwaremodulen, die sich von Teilen der Softwaremodule unterscheiden, die in einem speziellen Software-Bearbeitungsgerät gespeichert sind.

Es ermöglicht auch die Aktualisierung nicht nur von Software, sondern auch von Hardware. Insbesondere ist das Umschalten der Durchführung von Software zu einem anderen Software-Bearbeitungsgerät während der Hardware-Aktualisierung bei einem Software-Bearbeitungsgerät vorgesehen.

Zudem wird eine kombinierte Aktualisierung von Software und Hardware bei unterschiedlichen Software-Bearbeitungsgeräten dadurch geschaffen, daß zunächst die Hardwareteile verändert werden, und anschließend die Softwareteile angepaßt werden, und zwar durch Einsatz des erfindungsgemäßen Verfahrens. Hierbei müssen nicht alle Komponenten gleichzeitig angepaßt werden, und demnach besteht keine Anforderung dahingehend, das verteilte Rechensystem global neu zu starten.

#### Kurzbeschreibung der Zeichnung

Nachfolgend werden bevorzugte Ausführungsformen der vorliegenden Erfindung unter Bezug auf die beiliegende Zeichnung beschrieben; es zeigen:

**Fig. 1** ein Blockschaltbild des Software-Bearbeitungsgeräts gemäß der vorliegenden Erfindung;

**Fig. 2** ein Blockschaltbild der in **Fig. 1** gezeigten Aktualisierungs-Steuerung;

**Fig. 3** ein Diagramm des Zustandskopierverfahrens gemäß der vorliegenden Erfindung;

**Fig. 4** ein Flußdiagramm gemäß dem in **Fig. 3** gezeigten Zustandskopierverfahren;

**Fig. 5** ein Zustandsdiagramm zum Darstellen des Status einer Partition des Software-Bearbeitungsgeräts;

**Fig.** 6a einen parallelen synchronen Modus zum Durchführen von Software in beiden Partitionen gemäß dem in **Fig.** 3 gezeigten Schritt 1;

**Fig.** 6b einen Status beider Partitionen gemäß dem in **Fig.** 3 gezeigten Schritt 2;

**Fig.** 6c einen Status beider Partitionen gemäß dem in **Fig.** 3 gezeigten Schritt 3;

**Fig.** 6d einen Status beider Partitionen gemäß dem in **Fig.** 3 gezeigten Schritt 4;

**Fig.** 6e einen Status beider Partitionen gemäß dem in **Fig.** 3 gezeigten Schritt 5;

**Fig.** 7 die erfundungsgemäße Vorgehensweise zum Aktualisieren der Software in einem verteilten Rechensystem mit einem Remote-Prozessor mit Vorladefähigkeit;

**Fig.** 8 die Aktualisierung von Software in einem verteilten Rechensystem mit einem Remote-Prozessor ohne Einfluß auf die Kompatibilität der Schnittstelle zu diesem nach der Software-Aktualisierung;

**Fig.** 9 die Aktualisierung von Software in einem verteilten Rechensystem mit einem Remote-Prozessor mit einem Einfluß auf die Kompatibilität der Schnittstelle zu diesem nach der Software-Aktualisierung;

**Fig.** 10 die erfundungsgemäße Vorgehensweise der Aktualisierung von Hardware eines Hauptprozessors in einem verteilten Rechensystem;

**Fig.** 11 die erfundungsgemäße Vorgehensweise für die Aktualisierung von Hardware und Software in einem Remote-Prozessor eines verteilten Rechensystems ohne Einfluß auf die Kompatibilität der Schnittstelle hierzu nach der Aktualisierung; und

**Fig.** 12 die erfundungsgemäße Vorgehensweise für die Aktualisierung der Hardware und der Software in einem Remote-Prozessor eines verteilten Rechensystems mit Einfluß auf die Kompatibilität der Schnittstelle hierzu nach der Aktualisierung.

#### Beschreibung der bevorzugten Ausführungsformen

**Fig.** 1 zeigt ein Blockschaltbild einer Ausführungsform des Software-Bearbeitungsgeräts gemäß der vorliegenden Erfundung. Das Software-Bearbeitungsgerät gemäß der vorliegenden Erfundung enthält zwei Partitionen A und B. Für die Partition A ist eine erste Prozessoreinheit 4, eine erste Speicherpartition 6 und eine erste Übernahmeeinheit 8 vorgesehen. Die erste Speicherpartition ist in einen ersten Datenspeicherabschnitt 10 und einen ersten Software-Speicherabschnitt 12 unterteilt.

Ferner wird dieselbe Struktur für die Seite B gewählt, die eine zweite Prozessoreinheit 14, eine zweite Speicherpartition 16 und eine zweite Übernahmeeinheit 18 enthält. Wie bei der Seite A, ist die zweite Speicherpartition 16 in einen zweiten Datenspeicherabschnitt 20 und einen zweiten Software-Speicherabschnitt 22 unterteilt.

Wie in **Fig.** 1 gezeigt, ist zum Koordinieren der Aktualisierung der Software zwischen der Seite A und der Seite B oder umgekehrt zusätzlich eine Aktualisierungs-Steuereinheit 24 vorgesehen, die beide Prozessoreinheiten 4 und 14 steuert, sowie eine Übertragungseinheit 26 zum Koppeln der ersten Speicherpartition 6 mit der zweiten Speicherpartition 16.

Wie in **Fig.** 1 gezeigt, sind die erste und zweite Übernahmeeinheit 8 und 18 jeweils der ersten und zweiten Speicherpartition 6 und 16 zugeordnet, und zwar zum Durchführen von Vorgabeaktionen in dem Fall, in dem Daten im Zusammenhang mit der alten Software lediglich teilweise übertragen werden. Insbesondere betreffen derartige Vorgabeaktionen eine neue Software, die keine vollständige Eingabe von Daten erfordert. Sie besteht beispielsweise aus der

Initialisierung von Datenvariablen auf einem spezifischen Wert.

Wie oben dargelegt, ermöglicht dies eine Übertragung von Daten durch die Übertragungseinheit 26 in skalierbarer Weise, da nicht übertragene Daten jeweils durch die Übernahmeeinheit 8 und 18 initialisierbar sind. Weiterhin bewirkt die Übertragungseinheit 26 entweder ein unverändertes Kopieren von Daten oder eine Umsetzung derselben in eine neue Darstellung für die neue Software unter Steuerung der Aktualisierungs-Steuereinheit 24. Hier kann die Umsetzung der Daten parallel zu und ohne Störung des Abschnitts der alten Software in der Betriebspartition erfolgen. Weiterhin wiederholen die Aktualisierungs-Steuereinheit 24 und die Übertragungseinheit 26 die Datenübertragung in dem Fall, die vorab bereits übertragen wurden, und zwar bei gleichzeitiger Fortsetzung des Betriebs der alten Software in der Betriebspartition.

Zudem weist die Aktualisierungs-Steuereinheit 26 eine Fortführung der alten Software in der Betriebspartition in einem Fall an, in dem eine Fehlersituation vor dem Umschalten auftritt. Eine andere Option besteht in einer Rück-Umschaltung derart, daß die Partition mit der alten Software erneut die Betriebspartition in einem Fall wird, in dem ein Fehler während der Durchführung der neuen Software nach dem Umschalten auftritt.

Wie in **Fig.** 2 gezeigt, erzielt die Aktualisierungs-Steuereinheit eine Aktualisierung, die sich in bidirektionaler Weise durchführen läßt, derart, daß entweder die Speicherpartition 6 oder 16 als Betriebspartition während der Aktualisierung der anderen Partition 16, 6 als Standby-Partition dient, bei der die neue Software geladen wird. Bei diesem Aktualisierungsprozeß werden Daten von der Betriebspartition zu der Standby-Partition über die Übertragungseinheit 26 in skalierbarer Weise übertragen.

Zum Erzielen dieser Skalierbarkeit ist die in **Fig.** 1 gezeigte Aktualisierungs-Steuereinheit 24 wie in **Fig.** 2 aufgebaut. Die Aktualisierungs-Steuereinheit 24 enthält eine Zustandsvergleichseinheit 28, eine Übertragungssteuereinheit 30, eine Umschalteinheit 32, eine Speicherverwaltungseinheit 34 und eine Softwareladeeinheit 36. Die Zustandsvergleichseinheit 28 ermöglicht den Vergleich des Zustands von Daten und von Software in den beiden Speicherpartitionen 6 und 16. Ferner ist die Übertragungssteuereinheit 30 zum Erzielen einer skalierbar flexiblen Übertragung von Daten oder Software zwischen den beiden Speicherpartitionen 6 und 16 vorgesehen. Die Umschalteinheit 32 führt das Umschalten der Durchführung der Software zwischen der Seite A und der Seite B oder umgekehrt unmittelbar dann durch, wenn die Zustandsvergleichseinheit 28 denselben Zustand für die Betriebspartition und die Standby-Partition detektiert. Die Speicherverwaltungseinheit 34 ist zum Allozieren, Deallocieren oder Kompaktieren von Speicher entweder bei der Speicherpartition 6 oder der Speicherpartition 16 vorgesehen, und ebenfalls zum Festlegen von Referenzinformation hierin. Schließlich ist die Softwareladeeinheit 36 zum Laden neuer Software in den Softwarespeicherabschnitt 12, 22 jeder Partition 6, 16 vorgesehen.

Nach der Beschreibung der grundlegenden Struktur des Software-Bearbeitungsgeräts gemäß der vorliegenden Erfundung unter Bezug auf die **Fig.** 1 und 2 folgt nun die Beschreibung der Funktionalität dieser Komponenten sowie ihre Wechselwirkung unter Bezug auf die **Fig.** 3 bis 7. Obgleich im folgenden die Beschreibung der Aktualisierung der Software für die Seite B beschrieben wird, sollte es nicht als die vorliegende Erfundung einschränkend angesehen werden, die ebenfalls in die umgekehrte Richtung zu der Seite A durchführbar ist.

Die **Fig.** 3 zeigt die grundlegenden Schritte zum Durch-

führen des Zustandskopierverfahrens gemäß der vorliegenden Erfindung. Wie in **Fig. 3** gezeigt, führen in einem Schritt **1** beide Partitionen einen parallelen synchronen Modus mit der Durchführung beispielsweise derselben Software durch.

Ferner betrifft der in **Fig. 3** gezeigte Schritt **2** das Laden neuer Software in der Standby-Partition bei gleichzeitiger Fortsetzung der Durchführung der alten Software in der Betriebspartition. Ferner erfolgt im Schritt **3** das Kopieren von Daten von der Betriebspartition in die Standby-Partition. Wie im unteren Teil im Zusammenhang mit diesem Schritt **3** gezeigt, kann hiermit auch ein Umsetzen kopierter Daten in der Standby-Partition in eine Darstellung verbunden sein, die für die neue Software geeignet ist. Hierbei wird das Kopieren und Umsetzen von Daten parallel zu und ohne Störung der Durchführung der alten Software in der Betriebspartition durchgeführt. Weiterhin kann gemäß der vorliegenden Erfindung das Kopieren und Umsetzen von Daten durch speziell vorgesehene Software oder Hardware durchgeführt werden.

Wie in **Fig. 3** gezeigt, erfolgt im Schritt **4** eine Initialisierung, die ebenfalls parallel zu und ohne Störung der alten Software erfolgt, die in der Betriebspartition abläuft. Der Initialisierungsschritt wird entweder unmittelbar nach dem Laden der neuen Software in der Standby-Partition im Schritt **2** durchgeführt oder sobald wie möglich in dem Fall, in dem er von Daten abhängt, die von der alten Software im Schritt **3** kopiert werden.

Wie oben beschrieben, können Daten im Zusammenhang mit der alten Software lediglich teilweise übertragen werden. Spezielle Initialisierungsschritte werden vor oder unmittelbar nach dem Umschalten zum Durchführen von Vorgabeinitialisierungsschritten durchgeführt, die keine vollständige Eingabe der Daten der alten Software erfordern.

Wie in **Fig. 3** gezeigt, erfolgt unmittelbar nach Erzielen eines geeigneten Zustands in der Standby-Partition im Schritt **5** ein Umschalten zum Durchführen der neuen Software. Es ist zu erkennen, daß das Umschalten im Hinblick auf einzelne Softwaremodule unmittelbar nach dem Erzielen desselben Zustands für zugeordnete Softwaremodule in beiden Partitionen durchgeführt werden kann. Liegen Daten im Zusammenhang mit der alten Software vor, die zum Zeitpunkt des Umschaltens lediglich einer teilweisen Übertragung von Daten nicht übertragen sind, so können diese Daten immer noch, falls erforderlich, vor dem Start der neuen Software übertragen werden.

Wie in **Fig. 3** gezeigt, wird im Hinblick auf den Schritt **3** und den Schritt **4** der Kopierprozeß zwischen den beiden Speicherpartitionen auch während dem Initialisierungsschritt für die Standby-Partition fortgesetzt. Der Grund hierfür besteht darin, daß die alte Software fortlaufend während dem Aktualisierungsprozeß fortgesetzt wird und bereits vorher übertragene Daten überschreiben kann. Somit wird der Übertragungsprozeß wiederholt als Hintergrundprozeß solange fortgeführt, bis das Umschalten zu der neuen Software erfolgt, um den sie verändernden Zustand der Betriebspartition zu berücksichtigen. Dieser wiederholte Aktualisierungsprozeß kann parallel zu dem Initialisierungsschritt für die Standby-Partition durchgeführt werden.

**Fig. 4** zeigt ein Flußdiagramm gemäß dem unter Bezug auf die **Fig. 3** erörterten Aktualisierungsprozeß. Insbesondere ist ersichtlich, daß nach einem Schritt **1** und **2** zum Laden neuer Software und zum Initialisieren des Speichers hierfür ein Hintergrundprozeß fortlaufend solange durchgeführt wird, bis das Umschalten stattfindet. Hier ist zu erkennen, daß der Hintergrundprozeß auch durch Aufteilen in mehrere Hintergrundprozesse durchführbar ist. Wird der selbe Zustand für die alte und neue Software detektiert, so findet ein unmittelbares Umschalten, gefolgt von einer Ab-

frage, statt, um zu bestimmen, ob zu übertragende Daten noch vorliegen, was die wiederholte Durchführung der alten Software im Rahmen einer Schleife erfordert.

Im folgenden werden spezifische Beispiele für das Zustandskopierverfahren gemäß der vorliegenden Erfindung unter Bezug auf die **Fig. 5** und **6** beschrieben. Die **Fig. 5** zeigt die Darstellung des Zustands einer Speicherpartition unter Einsatz eines Zustandsgraphen, und die **Fig. 6a** bis **6b** zeigen die Modifikation eines derartigen Zustandsgraphen während des Ablaufs des Zustandskopierverfahrens.

Wie in **Fig. 5** gezeigt, wird ein Zustand einer Speicherpartition unter Einsatz eines Zustandsgraphen mit Knoten und Kanten dargestellt. Hier repräsentiert ein Knoten typischerweise unterschiedliche Datenzustände, und Kanten repräsentieren eine Übertragung zwischen unterschiedlichen Datenzuständen durch Ausführung von Softwaremodulen, die den Kanten zugeordnet sind. Ein Beispiel besteht darin, daß der oberste Knoten Eingabedaten betrifft, die durch ein Eingabedaten-Verarbeitungssystemmodul in für die weitere Verarbeitung geeignete Daten umzusetzen sind. Knoten mit eingefügten Kanten stellen die Wechselwirkung von zwei Softwaremodulen dar, bei denen die Ausgangsdaten eines Softwaremoduls Eingangsdaten des anderen Softwaremoduls darstellen und umgekehrt.

Wie in **Fig. 6** gezeigt, eignet sich diese Darstellung zum Erläutern der in **Fig. 3** gezeigten unterschiedlichen Schritte. Insbesondere betrifft **Fig. 6a** den parallelen synchronen Modus zum Durchführen derselben Software in der Betriebspartition und der Standby-Partition vor dem Start des Aktualisierungsprozesses. Wie in **Fig. 6b** gezeigt, wird während dem Laden der neuen Software im Schritt **2** die anhand der Kanten dargestellte Wechselwirkung unterschiedlicher Softwaremodule unterbrochen, und das Laden der neuen Software beginnt. Wie in **Fig. 6b** gezeigt, können Daten in unterschiedliche Kategorien aufgeteilt werden, wie bereits oben dargelegt. Hier bezeichnen die schwarzen Knoten Daten in der neuen Software, die identisch von der alten Software zu kopieren sind. Im Gegensatz hierzu bezeichnen weiße Knoten Daten der neuen Software, die in keiner Weise von Daten der alten Software abhängen. Ein typisches Beispiel wären Daten, die neu aufgrund der Modifikation von Datenstrukturen eingeführt werden. Eine andere Kategorie von Knoten, die schraffiert gezeigt ist, betrifft Daten, die zum Angleichen an die neue Software eine Umsetzung erfordern.

Eine weitere Differenzierung, die grau dargestellt ist, betrifft Daten, die lediglich eine Teilkopie oder -umsetzung, ausgehend von der alten Software, unter zusätzlichem Einsatz des Übernahmemechanismus erfordern, und zwar zum Reduzieren des an die neue Partition zu übertragenen Datenumfangs. Insgesamt wird, wie in **Fig. 6c** gezeigt, lediglich für die Daten der letzten drei Kategorien ein Kopier- und/oder Umsetzvorgang zwischen der Betriebspartition und der Standby-Partition durchgeführt.

Das Ergebnis des in **Fig. 3** gezeigten Schritts **4** ist in **Fig. 6d** gezeigt. Nach dem Initialisieren der neuen Software werden die Wechselbeziehungen der unterschiedlichen Datenkomponenten erneut eingeführt. Wie bereits oben unter Bezug auf die **Fig. 3** und **4** dargelegt, wird das Zustandskopierverfahren gemäß der vorliegenden Erfindung in dem Fall iteriert, in dem Daten durch die alte Software während des Aktualisierungsprozesses überschrieben werden. Somit zeigt die **Fig. 6d** die Situation vor dem Umschalten, bei der das Kopieren/Umsetzen auch nach der Initialisierung in Schritt **4** fortgeführt wird. Nach dem Umschalten im Schritt **5** existieren diese Kanten zum Darstellen des Kopierens/Umsetzens von Daten nicht mehr länger, wie in **Fig. 6e** gezeigt. Nach dem Umschalten entspricht der Status erneut dem oben beschriebenen, parallelen synchronen Modus.

Somit ist bei dem Zustandskopierverfahren der von der alten Software zu der neuen Software kopierte Status und gegebenenfalls der Gesamtzustand sowohl in der alten als auch in der neuen Version definiert. Prinzipiell kann der Betrieb mit jeder der Softwareversionen durchgeführt werden, da der Zustand für beide Versionen vollständig ist. Wichtig für das Zustandskopierverfahren ist die Tatsache, daß keine gleichzeitige Durchführung von Software in der Betriebsspartition und der Standby-Partition erfolgt, mit Ausnahme der Aktualisierungsfunktion selbst.

Gemäß dem erfundungsgemäßen Zustandskopierverfahren ist es auch möglich, den Aktualisierungsprozeß vor dem Umschalten im Fall des Auftretens einer Fehlersituation zu beenden und den Betrieb mit der alten Software fortzusetzen. Zudem ist das Durchführen einer Rückschaltung im Fall des Auftretens eines Fehlers während der Durchführung der neuen Software nach dem Umschalten so möglich, daß die alte Software erneut die Betriebsssoftware wird. Dieses Rückumschalten kann zu einer Datenübertragung mit einem Datenkopievorgang und der Umsetzung vom oben genannten Typ führen.

Während vorangehend das Zustandskopierverfahren im Hinblick auf ein Software-Bearbeitungsgerät beschrieben wurde, erfolgt nun die Beschreibung der Anwendung des Zustandskopierverfahrens auf ein verteiltes Rechensystem unter Bezug auf die **Fig. 7** bis **12**.

Wie in **Fig. 7** gezeigt, enthält das verteilte Rechensystem einen Hauptprozessor **38** und einen Remote-Prozessor **40**. Typischerweise weist der Hauptprozessor **38** die in **Fig. 1** gezeigte Struktur auf und ist in **Fig. 7** nur teilweise gezeigt. Ferner ist ein Remote-Prozessor **40** vorgesehen, der zumindest die Option zum Vorladen von Software in eine Speicherpartition **46** des Remote-Prozessors **40** bieten muß. Alternativ kann auch der Remote-Prozessor **40** die Struktur des erfundungsgemäßen Software-Bearbeitungsgeräts aufweisen, wie in **Fig. 9** gezeigt. Der Hauptprozessor **38** und der Remote-Prozessor **40** sind über eine Verbindungsleitung **42** verbunden. Jeder der Remote-Prozessor ist mit zumindest einer Aktualisierungsvorrichtung **44** zum Koordinieren der Aktualisierung in dem Remote-Prozessor unter Wechselwirkung mit dem Hauptprozessor **38** versehen.

Die **Fig. 7** zeigt im ersten Fall die Anwendung des erfundungsgemäßen Zustandskopierverfahrens in einem verteilten Rechensystem. Hier wird lediglich die Software in dem Remote-Prozessor **40** so aktualisiert, daß die neue Software anfänglich bei einer Speicherpartition **46** des Remote-Prozessors **40** geladen wird. Damit das Zustandskopierverfahren eingesetzt werden kann, muß der Remote-Prozessor **40** ein Vorladen so ermöglichen, daß das Bereitstellen von Diensten während dem Laden der neuen Software möglich ist, und daß nach dem Laden Daten, ausgehend von dem Hauptprozessor, aktualisiert werden können. Ist dies der Fall, so kann Software in dem Remote-Prozessor **40** ohne einen globalen Neustart des verteilten Rechensystems aktualisiert werden. Hierfür wird, sobald die neue Software in dem Remote-Prozessor **40** installiert ist, der Zustand der Speicherpartition **46** in dem Remote-Prozessor gemäß dem Zustand der Speicherpartition in dem Hauptprozessor **38** aktualisiert, bei gleichzeitigem Fortsetzen der Durchführung der Software in dem Hauptprozessor **38**. Schließlich wird die Durchführung der Software in dem Remote-Prozessor **40** zu der neuen Software unmittelbar dann umgeschaltet, wenn ein Abgleich zu dem Zustand des Hauptprozessors **38** erzielt ist.

Ferner kann bei dem Zustandskopierverfahren ein schnelles Aktualisieren des Remote-Prozessors **40** erforderlich sein, in Abhängigkeit davon, welche Art und wieviel Software zu aktualisieren ist. Hierbei bestehen in einem Fall, in

dem lediglich unkritische und/oder ein begrenzter Umfang der Software aktualisiert wird, keine hohen Anforderungen an die Aktualisierungsgeschwindigkeit. Demnach kann selbst bei Aktualisierung mehrerer Remote-Prozessoren eine Aktualisierungszeit erzielt werden, die konsistent zu der Unterbrechungszeit für einen Aktualisierungsprozeß ist.

Die **Fig. 8** zeigt einen weiteren Fall, gemäß dem Software nicht nur in dem Remote-Prozessor **40**, sondern auch in dem Hauptprozessor **38** aktualisiert wird, und bei dem der Aktualisierungsprozeß einen Einfluß auf die Kompatibilität der Schnittstelle hat. Hier wird die Softwareaktualisierung in zwei Schritten durchgeführt, indem zunächst die Software in dem Remote-Prozessor **40**, wie oben dargelegt, aktualisiert wird und anschließend die Software in dem Hauptprozessor **38** unter Einsatz des oben beschriebenen Zustandskopierverfahrens aktualisiert wird. Werden nicht alle Remote-Prozessoren in dem verteilten Rechensystem zur gleichen Zeit aktualisiert, so besteht keine Anforderung für einen globalen Neustart des Systems.

Die **Fig. 9** betrifft denselben Fall, wie den in **Fig. 8** gezeigten, mit dem Unterschied, daß nach der Aktualisierung der Software in dem Hauptprozessor **38** und dem Remote-Prozessor **40** die Schnittstelle hierzwischen inkompatibel wird.

In diesem Fall sollte auch der Remote-Prozessor **40** die oben unter Bezug auf die **Fig. 1** beschriebene Struktur aufweisen, so daß eine gleichzeitige Aktualisierung von Software in dem Remote-Prozessor **40** und dem Hauptprozessor **38** bei hierzwischen modifizierter Schnittstelle durch gleichzeitiges Durchführen des erfundungsgemäßen Zustandskopierverfahrens jeweils in dem Hauptprozessor und dem Remote-Prozessor **40** erzielt wird.

Hierbei sollte in einem Fall, in dem nicht kritische Teile des verteilten Rechensystems betroffen sind, das Zustandskopierverfahren eingesetzt werden, indem der veränderte Teil des Systems ausgekoppelt wird, anschließend gleichzeitig die Software aktualisiert wird, schließlich die veränderten Teile des verteilten Rechensystems wieder eingekoppelt werden. Sind Daten von der alten Software zu der neuen Software übertragen, so sollte das Kopieren/Umsetzen vor dem Start und dem Einkoppeln der neuen Software durchgeführt werden. Sind andererseits kritische Teile während der Aktualisierung der Software betroffen, so sollte der Remote-Prozessor **40** vorab mit der neuen Software geladen werden, um eine zu lange Unterbrechungszeit des verteilten Rechensystems während des Aktualisierungsprozesses zu vermeiden.

Weitere Möglichkeiten bestehen darin, daß die neue Software in dem Remote-Prozessor **40** mit Daten von dem Hauptprozessor **38** aktualisiert wird. Zudem können Funktionen zum Unterstützen der Übertragung von Daten von der alten zu der neuen Software bei dem Remote-Prozessor eingeführt werden.

Nach der Beschreibung der Aktualisierung von Software in unterschiedlichen Systemkonfigurationen unter Einsatz des erfundungsgemäßen Zustandskopierverfahrens wird nun ein kombiniertes Aktualisieren von Hardware und Software unter Bezug auf die **Fig. 10** bis **12** beschrieben.

Die **Fig. 10** betrifft die Aktualisierung von Hardware des Hauptprozessors **38**. Typischerweise werden Hardwarekomponenten ausgetauscht, indem die auszuwechselnden Hardwarekomponenten ausgekoppelt werden, anschließend diese ersetzt werden und schließlich ein erneutes Einkoppeln derselben durchgeführt wird.

Die **Fig. 11** zeigt den nächsten Fall, bei dem Software sowohl in dem Remote-Prozessor **40** als auch dem Hauptprozessor **38** ohne Einfluß auf die Kompatibilität der Schnittstelle aktualisiert wird. Ferner wird in dem in **Fig. 11** gezeigt

ten Fall auch dem Remote-Prozessor **40** zugeordnete Hardware ausgewechselt. Hierfür werden auszutauschende Komponenten, die dem Remote-Prozessor **40** zugeordnet sind, zunächst unter Einsatz unter Bezug auf die in **Fig. 10** beschriebene Vorgehensweise ausgewechselt. Anschließend wird die Software in dem Remote-Prozessor **40** und dem Hauptprozessor **38** unter Einsatz der mit Bezug auf die **Fig. 8** beschriebene Vorgehensweise aktualisiert.

**Fig. 12** zeigt einen weiteren Fall für die Anwendung des Zustandskopierverfahrens, bei dem dem Remote-Prozessor zugeordnete Hardwarekomponenten gleichzeitig mit der Aktualisierung der Software des Remote-Prozessors und des Haupt-Prozessors **38** ausgewechselt werden, derart, daß eine Inkompatibilität der Software nach der Aktualisierung entsteht. Führt die Veränderung der Hardware und Software im Hinblick auf den Remote-Prozessor **40** zu einer Inkompatibilität innerhalb des Remote-Prozessors und im Hinblick auf die neuen Hardware- und Softwarekomponenten, so wird zunächst die Hardware bei dem Remote-Prozessor **40** ausgewechselt. Anschließend wird die Aktualisierung der Software, wie oben unter Bezug auf die **Fig. 9** beschrieben, durchgeführt.

Im Gegensatz hierzu ist die Situation komplizierter, wenn auch das Auswechseln der Hardwarekomponenten bei dem Remote-Prozessor **40** zu einer Inkompatibilität im Hinblick auf die aktualisierte Software bei dem Remote-Prozessor führt. Ist die Veränderung der Hardware und der Software im Hinblick auf den Leistungsumfang des verteilten Rechensystems unkritisch, so kann dieselbe Vorgehensweise eingesetzt werden, wie sie unter Bezug auf die **Fig. 11** beschrieben wurde.

Ist diese Hardwareveränderung jedoch kritisch, so sollten die jeweiligen Hardwarekomponenten dupliziert bei dem Remote-Prozessor **40** vorgesehen sein, und auch die Software sollte entweder gemäß der **Fig. 7** und **8** bei dem Remote-Prozessor **40** vorab geladen sein, oder der Remote-Prozessor **40** sollte zwei Partitionen aufweisen. Eine weitere Anforderung für diesen Fall besteht darin, daß die Bearbeitungsgeschwindigkeit des Remote-Prozessors **40** hoch genug ist. Sind diese Bedingungen erfüllt, so ist es möglich, die kombinierte Aktualisierung ohne übermäßige Systemunterbrechung durchzuführen.

#### Bezugszeichenliste

<b>2</b>	Software-Bearbeitungsgerät	45
<b>4</b>	Prozessoreinheit der Seite A	
<b>6</b>	Speicherpartition der Seite A	
<b>8</b>	Übernahmeeinheit der Seite A	
<b>10</b>	Datenspeicherabschnitt der Seite A und Speicherpartition der Seite A	50
<b>12</b>	Softwarespeicherabschnitt der Seite A und Speicherpartition der Seite A	
<b>14</b>	Prozessoreinheit der Seite B	
<b>16</b>	Speicherpartition der Seite B	55
<b>18</b>	Übernahmeeinheit der Seite B	
<b>20</b>	Datenspeicherabschnitt der Seite B und Speicherpartition der Seite B	
<b>22</b>	Softwarespeicherabschnitt der Seite B und Speicherpartition der Seite B	60
<b>24</b>	Aktualisierungs-Steuereinheit	
<b>26</b>	Übertragungseinheit	
<b>28</b>	Zustandsvergleichseinheit	
<b>30</b>	Übertragungssteuereinheit	
<b>32</b>	Umschalteinheit	
<b>34</b>	Speicherverwaltungseinheit	
<b>36</b>	Softwareladeeinheit	
<b>38</b>	Hauptprozessor	

**40** Remote-Prozessor  
**42** Verbindungsleitung  
**44** Aktualisierungsvorrichtung im Remote-Prozessor  
**46** Speicherpartition des Remote-Prozessors

5

#### Patentansprüche

1. Software-Bearbeitungsgerät vom Typ mit Aktualisierungs-Funktionalität, enthaltend:
  - a) eine Speichervorrichtung (**6, 16**), unterteilt in
    - a1) eine ausführende Speicherpartitionierungsvorrichtung (**6**) zum Speichern einer ersten Gruppe von Softwaremodulen und zugeordneter Daten, und
    - a2) eine Standby-Speicherpartitionivorrichtung (**16**) zum Speichern einer zweiten Gruppe von Softwaremodulen und zugeordneter Daten,
  - b) eine Aktualisierungs-Steuervorrichtung (**24**) zum Aktualisieren eines Zustands der neuen Software in der Standby-Speicherpartitionivorrichtung (**16**) gemäß dem Zustand der alten Software in der ausführenden Speicherpartitionivorrichtung (**6**) während der Fortführung der Durchführung der alten Software, und
  - c) eine Übertragungsvorrichtung (**26**) für die skalierbare Übertragung von Daten von der ausführenden Speicherpartitionivorrichtung (**6**) zu der Standby-Speicherpartitionivorrichtung (**16**).
2. Software-Bearbeitungsgerät nach Anspruch 1, dadurch gekennzeichnet, daß die Aktualisierungs-Steuervorrichtung (**24**) enthält:
  - d) eine Speicherverwaltungsvorrichtung (**34**) zum Allokieren und Deallokieren von Speicherabschnitten für neue und alte Software und die Daten sowie zum Verwalten von Referenzinformation hierfür, und
  - e) eine Übertragungssteuereinheit (**30**) zum Steuern der Übertragungsvorrichtung (**26**) gemäß den Befehlen für die skalierbare Übertragung von Daten.
3. Software-Bearbeitungsgerät nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß die Aktualisierungs-Steuervorrichtung (**24**) ferner eine Umschaltvorrichtung (**32**) enthält, sowie eine Zustands-Vergleichsvorrichtung (**28**) für das unmittelbare Umschalten zu der Durchführung der neuen Software, sobald derselbe Zustand für die Standby-Speicherpartitionivorrichtung (**16**) und die ausführende Speicherpartitionivorrichtung (**6**) durch die Zustands-Vergleichsvorrichtung (**28**) detektiert ist.
4. Software-Bearbeitungsgerät nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß jeder Speicherpartitionivorrichtung (**6, 16**) zumindest eine Übernahmeverrichtung (**8, 18**) zugeordnet ist, und zwar zum Ausführen von Vorgabeaktionen für den Fall, daß Daten im Zusammenhang mit der alten Software lediglich zum Teil übertragen wird, derart, daß die Übernahmeverrichtung (**8, 18**) unmittelbar nach dem Umschalten aktiviert ist.
5. Software-Bearbeitungsgerät nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß die Übertragungsvorrichtung (**16**) entweder Daten unverändert kopiert oder nach einer Umsetzung in eine neue Darstellung für die neue Software überträgt.
6. Software-Bearbeitungsgerät nach Anspruch 5, dadurch gekennzeichnet, daß die Übertragungsvorrichtung (**26**) die Umsetzung von Daten parallel zu und

ohne Störung der Durchführung der alten Software in der ausführenden Speicherpartitioniervorrichtung (6) durchführt.

7. Software-Bearbeitungsgerät nach Anspruch 5 oder 6, dadurch gekennzeichnet, daß die Übertragungsvorrichtung (26) eine ausgewiesene Umsetzvorrichtung enthält.

8. Software-Bearbeitungsgerät nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, daß die Aktualisierungs-Steuervorrichtung (24) wiederholt den Aktualisierungsprozeß solange durchführt, bis die Umschaltvorrichtung (32) zu der Durchführung der neuen Software umschaltet, und zwar zum Verfolgen des sich verändernden Zustands in der ausführenden Speicherpartitioniervorrichtung (6).

9. Software-Bearbeitungsgerät nach einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, daß bei Vorliegen von Daten im Zusammenhang mit der alten Software, die zum Zeitpunkt des Umschaltens nicht übertragen sind, die Übertragungsvorrichtung (26), falls erforderlich, diese Daten vor dem Start der neuen Software überträgt.

10. Software-Bearbeitungsgerät nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, daß die Aktualisierungs-Steuervorrichtung (24) die Fortführung der alten Software in der ausführenden Speicherpartitioniervorrichtung in dem Fall anweist, in dem eine Fehlersituation vor dem Umschalten auftritt.

11. Software-Bearbeitungsgerät nach einem der Ansprüche 1 bis 10, dadurch gekennzeichnet, daß die Umschaltvorrichtung (32) einen Rückschaltvorgang so durchführt, daß die Partitionierung mit der alten Software erneut die ausführende Speicherpartitioniervorrichtung (6) wird, und zwar in dem Fall, in dem ein Fehler während der Durchführung der neuen Software nach dem Umschalten auftritt.

12. Verteiltes Rechensystem vom Typ mit Aktualisierungs-Funktionalität, enthaltend:

- a) zumindest eine Hauptprozessorenvorrichtung (38), ausgewählt aus mehreren Prozessoren des verteilten Rechensystems zum Handhaben der Verteilung der Prozeßteilaufgaben in dem verteilten Rechensystem sowie der Interaktionen der in diesem enthaltenden Prozessoreinheiten,
- b) zumindest eine Remote-Prozessorenvorrichtung (40) mit einer Aktualisierungsvorrichtung (44) zum Aktualisieren neuer Software in einer Speicherpartition (46) der Remote-Prozessorenvorrichtung (40) derart, daß ein Zustand der neuen Software an einen Zustand der Hauptprozessorenvorrichtung (38) angeglichen ist und die Durchführung der Software in der Remote-Prozessorenvorrichtung zu der neuen Software umgeschaltet ist, sobald der Abgleich erzielt ist.

13. Verteiltes Rechensystem nach Anspruch 12, dadurch gekennzeichnet, daß für den Fall einer nach der Aktualisierung der neuen Software in der Remote-Prozessorenvorrichtung (40) immer noch kompatiblen Schnittstelle zwischen der Remote-Prozessorenvorrichtung (40) und der Hauptprozessorenvorrichtung (48) die Hauptprozessorenvorrichtung (38) gemäß einem der Ansprüche 1 bis 11 implementiert ist, zum Erzielen einer kombinierten Aktualisierung von Software in der Remote-Prozessorenvorrichtung (38) und der Hauptprozessorenvorrichtung (40).

14. Verteiltes Rechensystem nach Anspruch 12, dadurch gekennzeichnet, daß im Fall einer nach der Softwareaktualisierung in der Hauptprozessorenvorrichtung

5

10

15

20

25

30

35

40

45

50

55

60

65

(38) und der Remote-Prozessorenvorrichtung (40) nicht mehr kompatiblen Schnittstelle zwischen der Remote-Prozessorenvorrichtung (40) und der Hauptprozessorenvorrichtung (38) die Remote-Prozessorenvorrichtung (40) nach einem der Ansprüche 1 bis 11 implementiert ist und gleichzeitig die Softwareaktualisierung zum Anpassen an die modifizierte Schnittstelle durchführt.

15. Zustandskopierverfahren für ein Rechensystem mit zumindest zwei logischen Partitionen, enthaltend die Schritte:

- a) Aktualisieren eines Zustands der neuen Software in einer Standby-Partitioniervorrichtung (16) an den Zustand der alten Software in einer ausführenden Partitioniervorrichtung bei Fortführung der Durchführung der alten Software,
- b) Umschalten zum Durchführen der neuen Software, sobald derselbe Zustand für die Standby-Partitioniervorrichtung (16) und die ausführende Partitioniervorrichtung (6) erzielt wird.

16. Zustandskopierverfahren nach Anspruch 15, dadurch gekennzeichnet, daß der Aktualisierschritt a) unterteilt ist gemäß:

- c) Laden der neuen Software in die Standby-Partitioniervorrichtung (16), und
- d) skalierbares Übertragen von Daten von der ausführenden Partitioniervorrichtung (6) zu der Standby-Partitioniervorrichtung (16).

17. Zustandskopierverfahren nach Anspruch 15 oder 16, dadurch gekennzeichnet, daß die Übertragung von Daten von der ausführenden Partitioniervorrichtung (6) zu der Standby-Partitioniervorrichtung (16) unterteilt ist gemäß:

- e) unverändertes Kopieren von zu übertragenden Daten, und
- f) Umsetzen von umzusetzenden Daten in eine neue Darstellung für die neue Software.

18. Zustandskopierverfahren nach Anspruch 17, dadurch gekennzeichnet, daß die Umsetzung von Daten parallel zu und ohne Störung der Durchführung der alten Software in der ausführenden Partitioniervorrichtung (6) durchgeführt wird.

19. Zustandskopierverfahren nach Anspruch 17 oder 18, dadurch gekennzeichnet, daß die Umsetzung von Daten durch einen ausgewiesenen Umsetschritt durchgeführt wird.

20. Zustandskopierverfahren nach einem der Ansprüche 15 bis 19, dadurch gekennzeichnet, daß der Aktualisierungsschritt a) auch einen Initialisierungsteilschritt aufweist, der parallel zu und ohne Störung der in der ausführenden Partitioniervorrichtung (6) ablaufenden alten Software durchgeführt wird.

21. Zustandskopierverfahren nach Anspruch 20, dadurch gekennzeichnet, daß der Initialisierungsschritt entweder unmittelbar nach dem Laden der neuen Software in die Standby-Partitioniervorrichtung (16) durchgeführt wird oder sobald wie möglich, falls er Daten der alten Software erfordert.

22. Zustandskopierverfahren nach einem der Ansprüche 15 bis 21, dadurch gekennzeichnet, daß der Aktualisierungsschritt a) wiederholt als Hintergrundprozeß solange durchgeführt wird, bis das Umschalten zu der neuen Software erfolgt, und zwar zum Verfolgen des sich verändernden Zustands in der ausführenden Partitioniervorrichtung (6).

23. Zustandskopierverfahren nach Anspruch 20 oder 21, dadurch gekennzeichnet, daß der Aktualisierungsschritt a) wiederholt parallel zu dem Initialisierungsschritt durchgeführt wird.

24. Zustandskopierverfahren nach einem der Ansprüche 15 bis 23, dadurch gekennzeichnet, daß für den Fall von im Zeitpunkt des Umschaltens nicht übertragener Daten im Zusammenhang mit der alten Software diese Daten, falls erforderlich, vor dem Start der neuen Software übertragen werden. 5

25. Zustandskopierverfahren nach einem der Ansprüche 16 bis 24, dadurch gekennzeichnet, daß im Teilschritt d) Daten im Zusammenhang mit der alten Software lediglich teilweise übertragen werden und daß ein spezieller Übernahmeschritt unmittelbar nach dem Umschalten zum Durchführen von Vorgabeaktionen ohne vollständige Eingabe von Daten durchgeführt wird. 10

26. Zustandskopierverfahren nach einem der Ansprüche 15 bis 25, dadurch gekennzeichnet, daß im Fall eines Fehlers vor dem Umschalten die Aktualisierung abgeschlossen wird und die Durchführung der alten Software in der ausführenden Partitionenvorrichtung (6) fortgeführt wird. 15

27. Zustandskopierverfahren nach einem der Ansprüche 15 bis 26, dadurch gekennzeichnet, daß ein Rück-Umschaltsschritt so durchgeführt wird, daß die ausführende Partitionenvorrichtung (6) mit der alten Software erneut die tatsächlich ausführende Partitionenvorrichtung (6) wird, und zwar im Fall eines Fehlers während der Durchführung der neuen Software nach dem Umschalten. 20

28. Zustandskopierverfahren nach Anspruch 27, dadurch gekennzeichnet, daß beim Rück-Umschalten eines Datenübertragung mit Datenkopieren und Umsetzen, falls erforderlich, durchgeführt wird, und zwar mit begrenzter oder ohne jegliche Störung. 30

29. Zustandskopierverfahren nach Anspruch 27 oder 28, dadurch gekennzeichnet, daß beim Rück-Umschalten ein Wiederherstellsschritt durchgeführt wird, der vor dem erneuten Start der alten Software durchgeführt wird. 35

30. Zustandskopierverfahren für ein verteiltes Rechensystem mit einer Hauptprozessorenvorrichtung (38) und zummindest einer Remote-Prozessorenvorrichtung (40), enthaltend die Schritte: 40

- a) Aktualisieren der neuen Software in einer ersten Speicherpartitionenvorrichtung (46) der Remote-Prozessorenvorrichtung (40), 45
- b) Aktualisieren eines Zustands der neuen Software zum Erzielen eines Abgleichs mit dem Zustand der Haupt-Prozessorenvorrichtung (38) bei Fortführung der Durchführung der Software in der Hauptprozessorenvorrichtung (38), und 50
- c) Umschalten der Durchführung der Software in der Remote-Prozessorenvorrichtung (40) zu der neuen Software, sobald ein Abgleich mit dem Zustand der Haupt-Prozessorenvorrichtung (38) erzielt ist. 55

31. Zustandskopierverfahren nach Anspruch 30, dadurch gekennzeichnet, daß bei nach der Aktualisierung der neuen Software in der Remote-Prozessorenvorrichtung (40) kompatibel bleibenden Schnittstelle zwischen der Remote-Prozessorenvorrichtung (40) und der Hauptprozessorenvorrichtung (38) eine kombinierte Aktualisierung von Software in der Remote-Prozessorenvorrichtung (40) und der Hauptprozessorenvorrichtung (38) durchgeführt wird, durch zusätzliches Ausführen des Zustandskopierverfahrens nach einem der Ansprüche 60 bis 29 in der Hauptprozessorenvorrichtung (38). 65

32. Zustandskopierverfahren nach Anspruch 30, dadurch gekennzeichnet, daß ein gleichzeitiges Aktuali-

sieren von Software in der Remote-Prozessorenvorrichtung (40) und der Hauptprozessorenvorrichtung (38) mit modifizierter Schnittstelle durch gleichzeitiges Durchführen des Zustandskopierverfahrens nach einem der Ansprüche 15 bis 29 in der Hauptprozessorenvorrichtung (38) und der Remote-Prozessorenvorrichtung (40) durchgeführt wird.

33. Zustandskopierverfahren nach einem der Ansprüche 30 bis 32, dadurch gekennzeichnet, daß ferner an die Remote-Prozessorenvorrichtung (40) angeschlossene Hardware-Komponenten ausgetauscht werden, und zwar durch Blockieren der auszutauschenden Hardwarekomponenten, anschließendes Ersetzen derselben und schließlich Freischalten derselben.

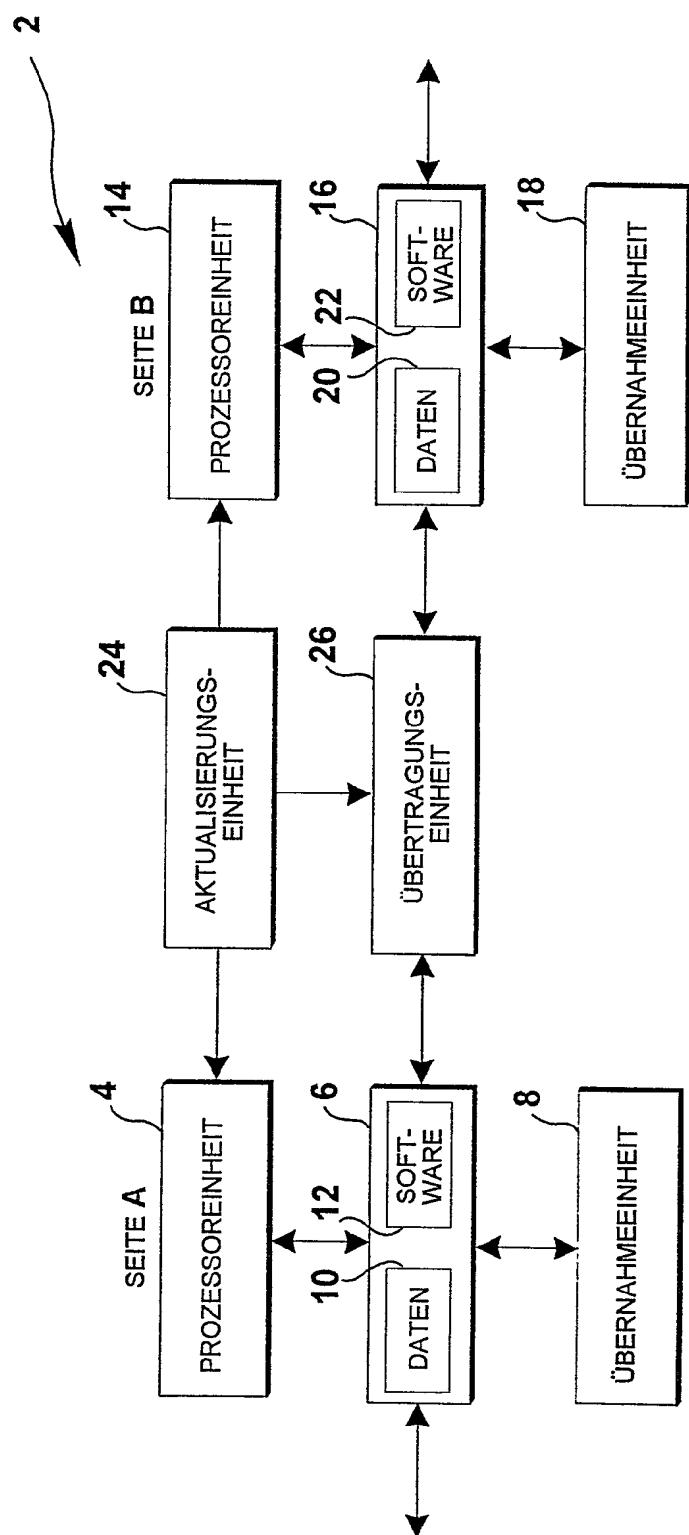
---

Hierzu 10 Seite(n) Zeichnungen

---

**- Leerseite -**

FIG.1



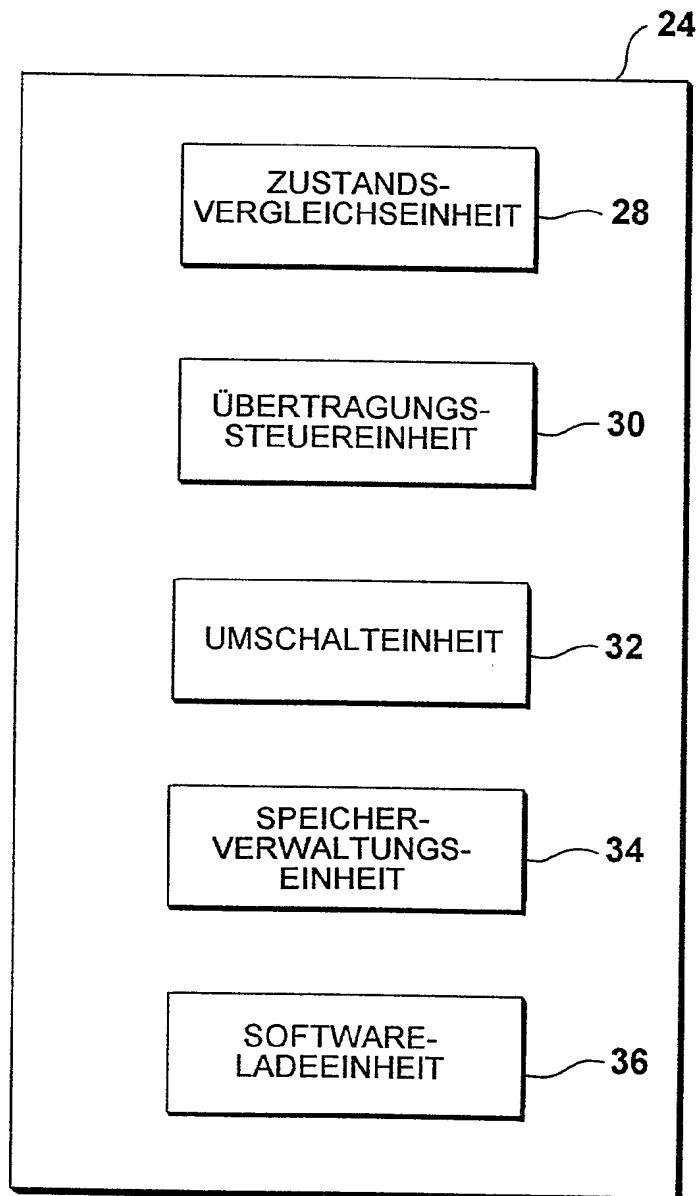
***FIG.2***

FIG.3

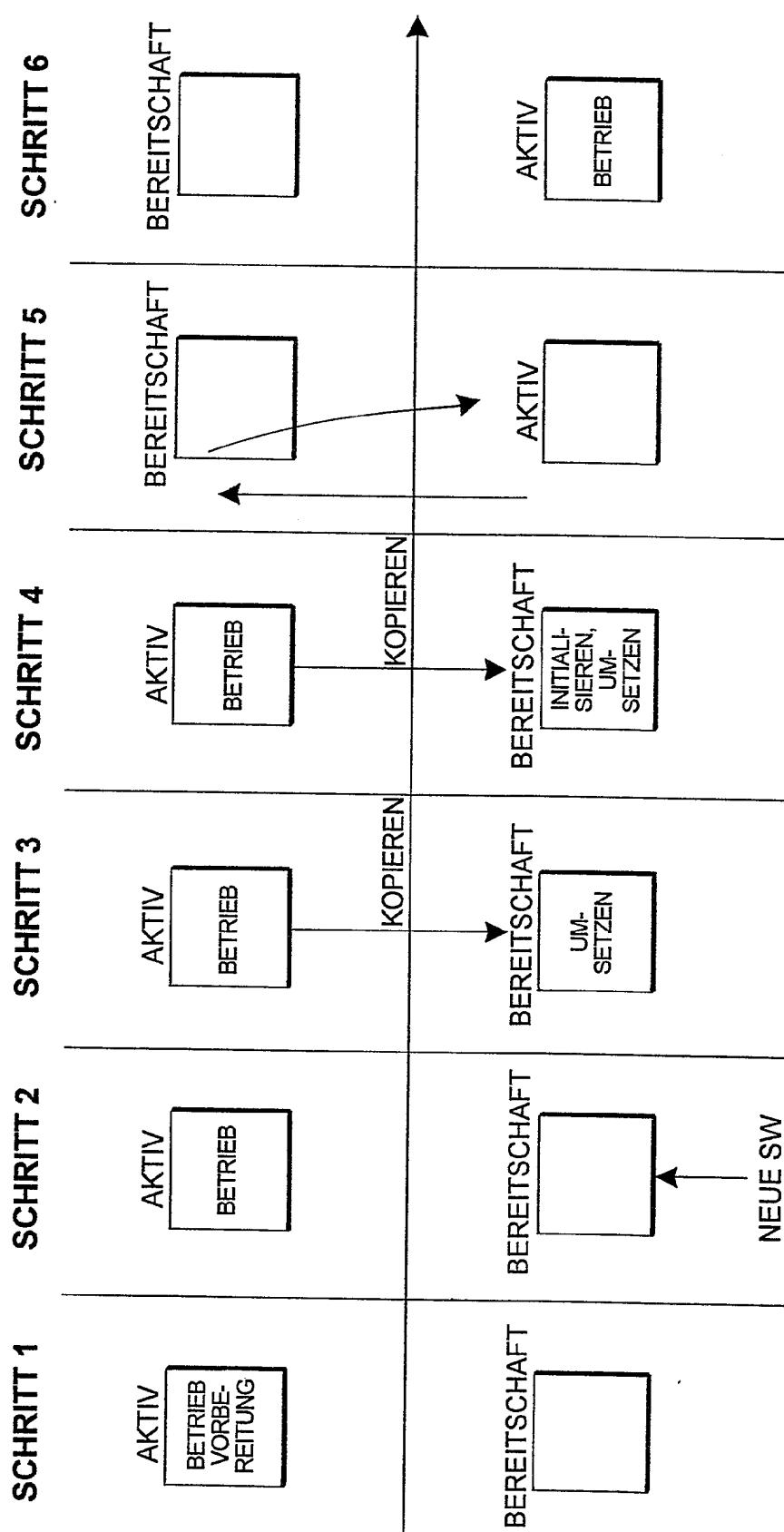
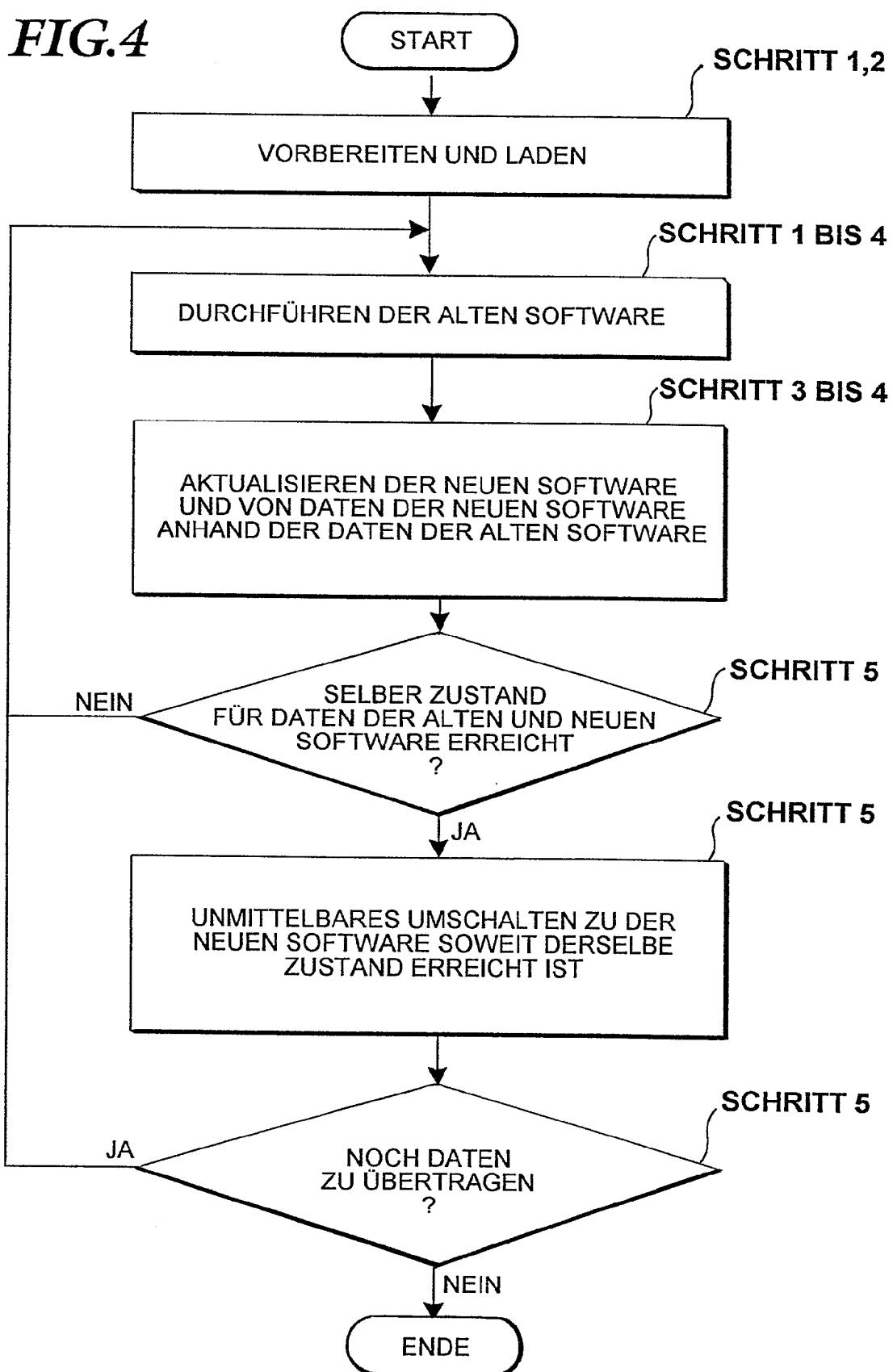
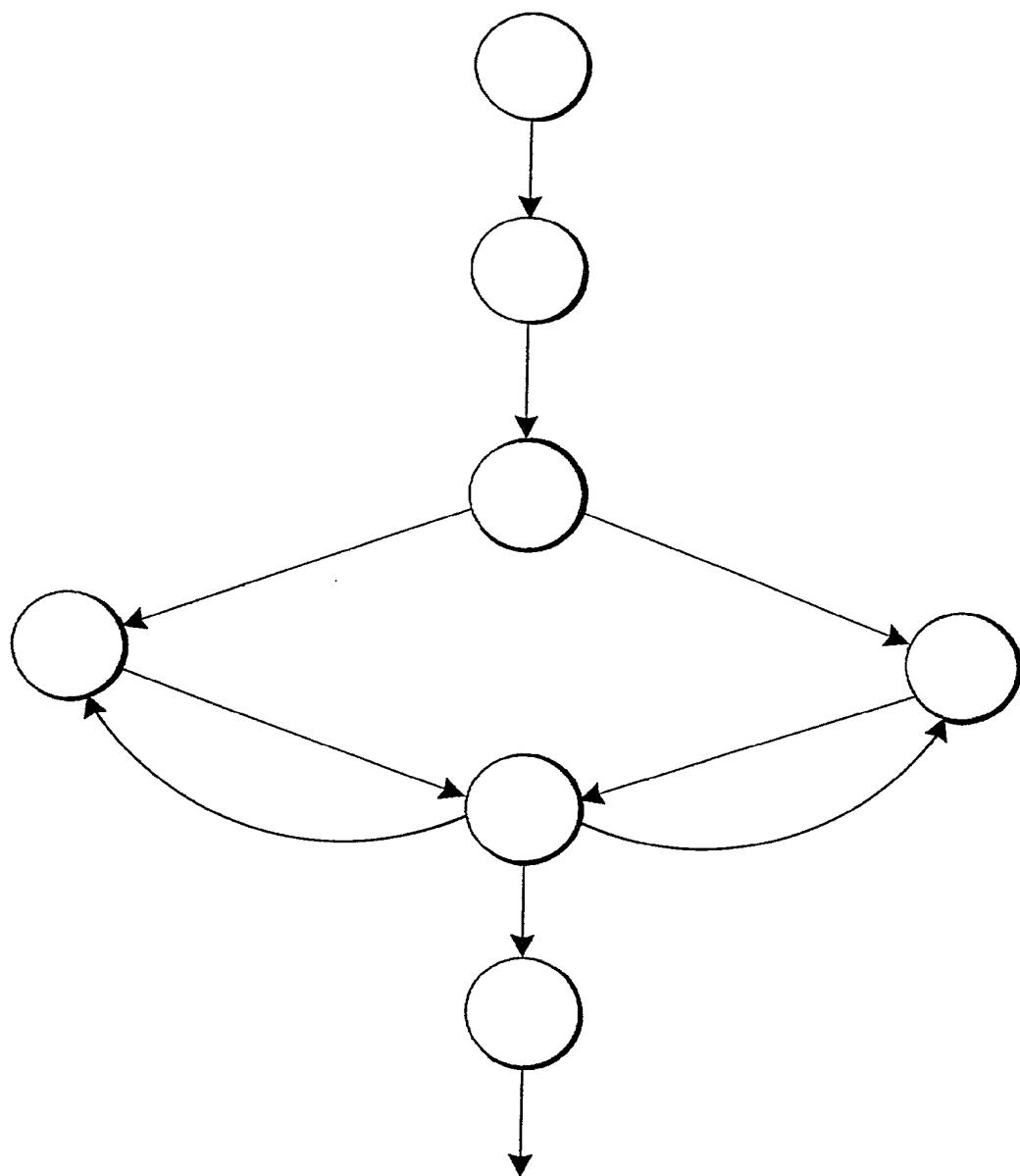


FIG.4



**FIG.5**

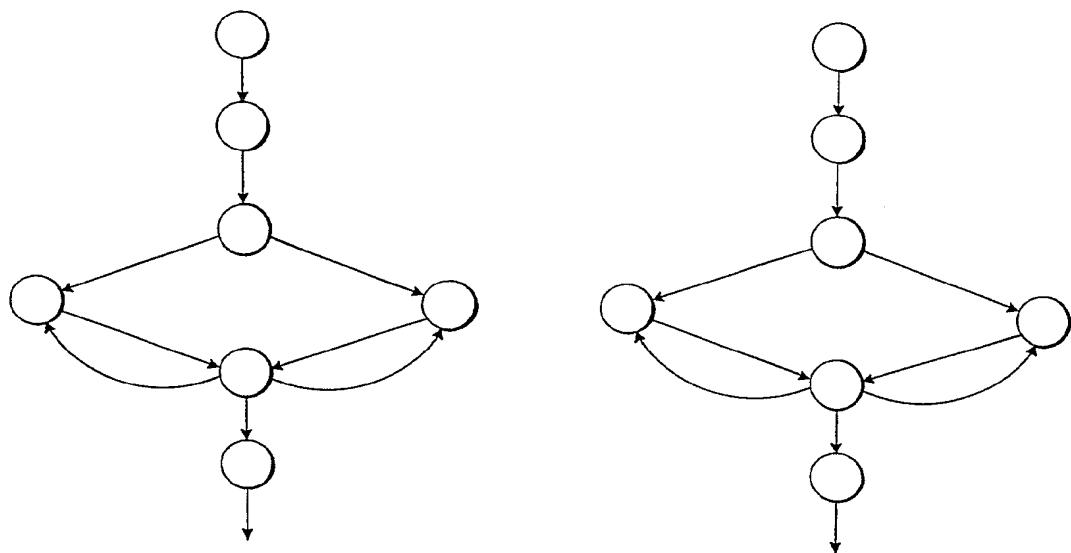
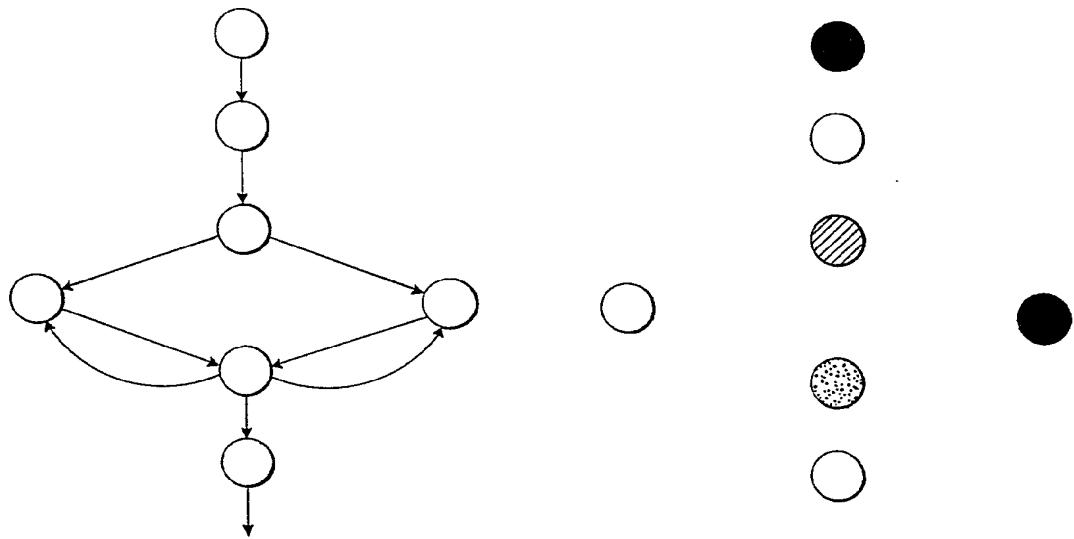
***FIG.6(a)******FIG.6(b)***

FIG. 6(c)

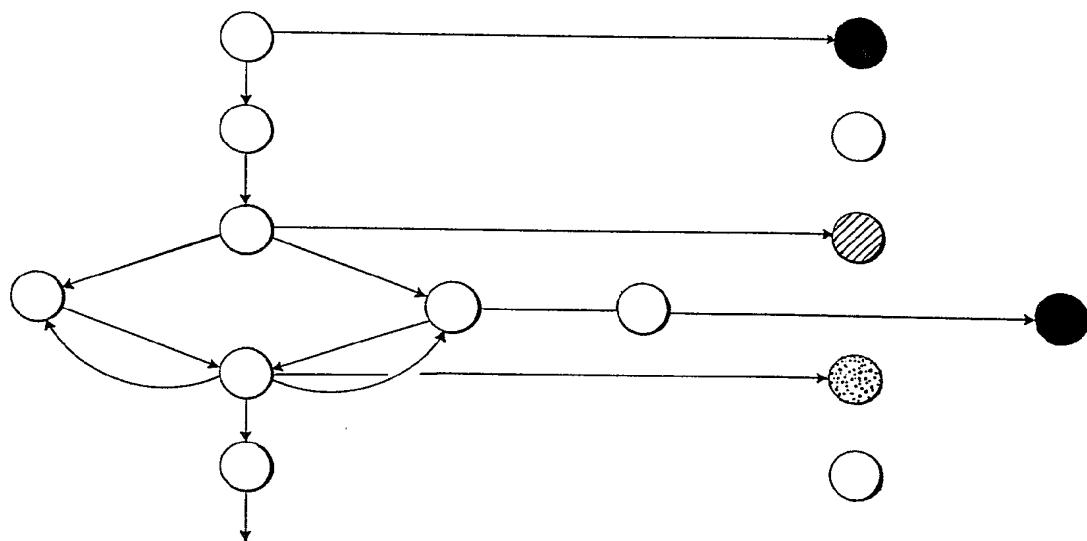


FIG. 6(d)

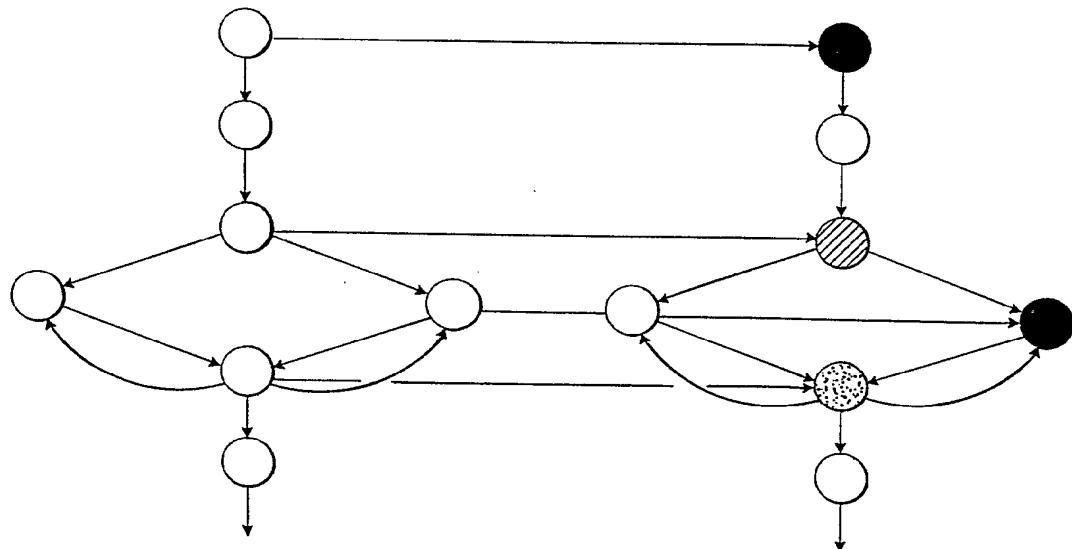


FIG.6(e)

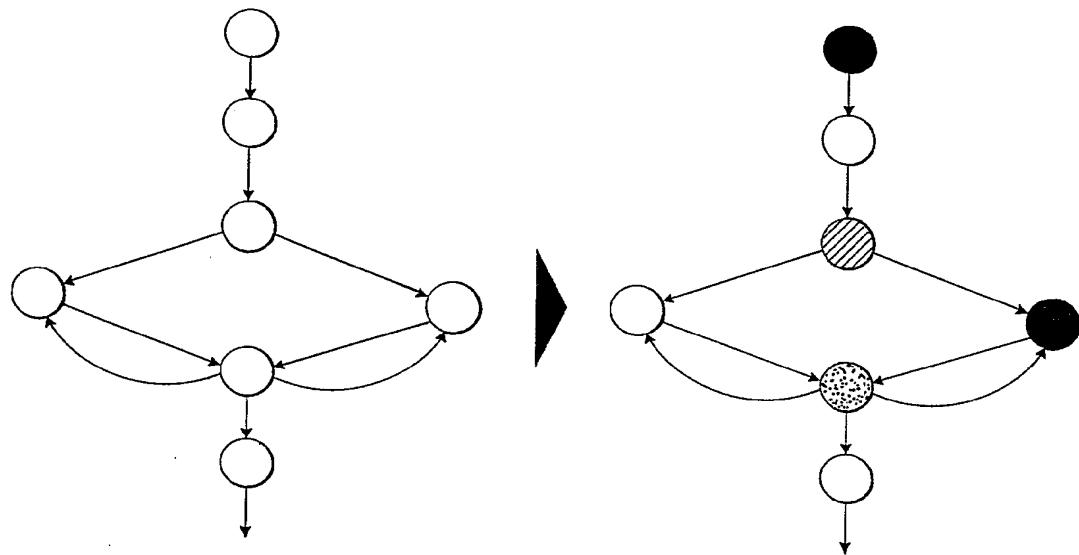


FIG.7

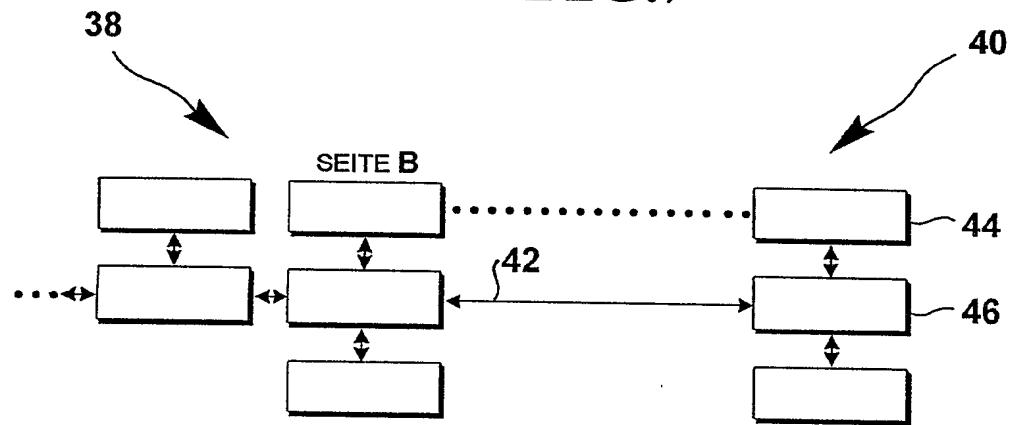
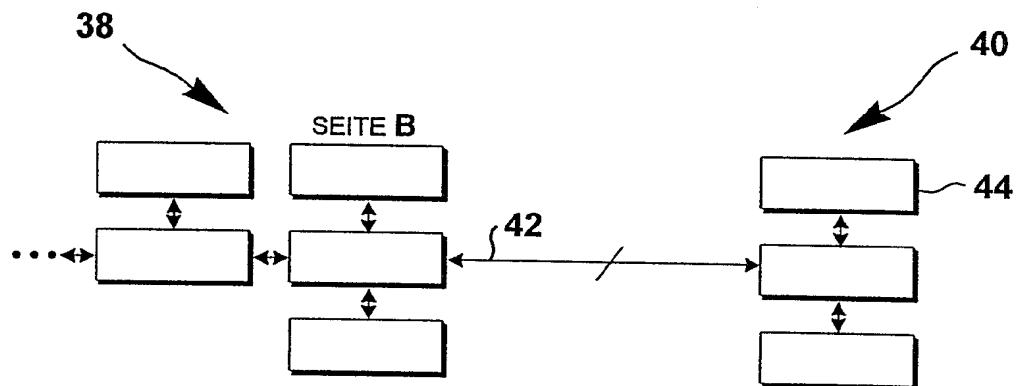


FIG.8



38

FIG.9

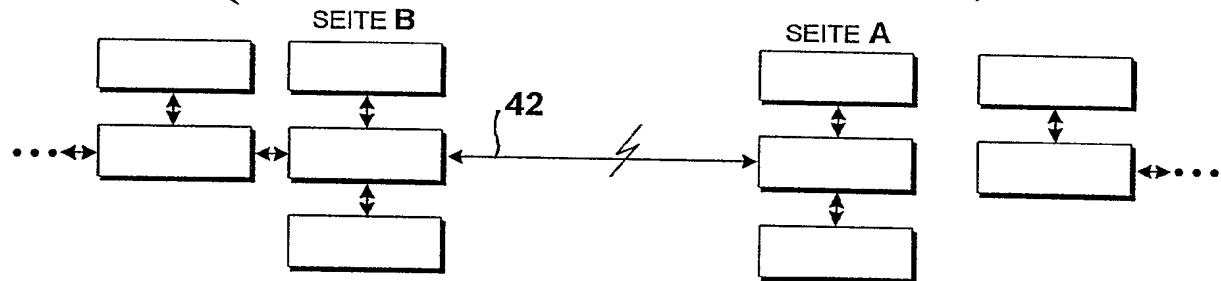


FIG.10

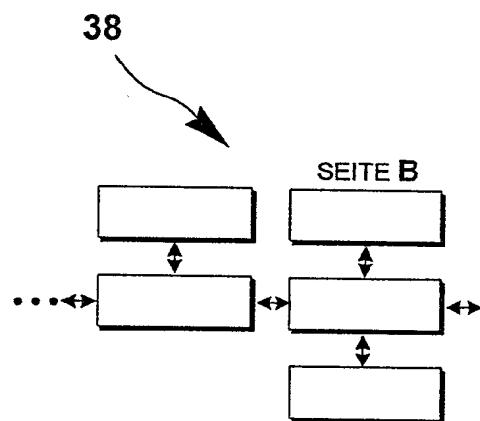
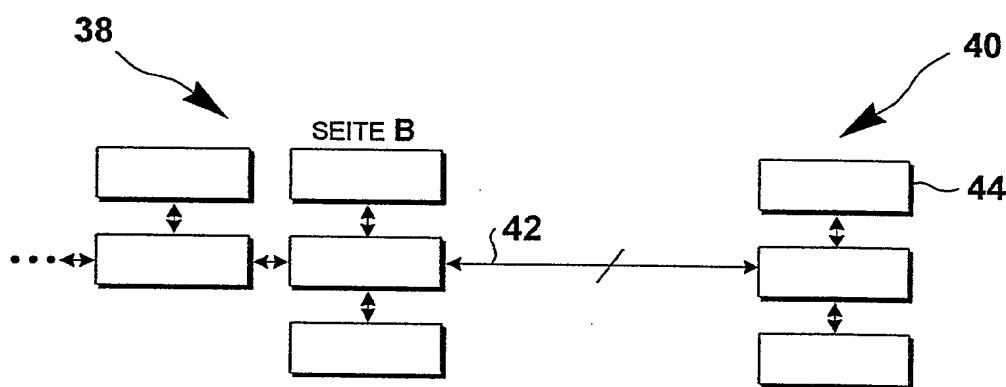


FIG.11



38

FIG.12

